Analysis of Unicorn Startups

Contents

1 Setup

1.1 Import Packages

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import seaborn as sns
```

2 Data Preparation

2.1 Load Data

```
pd.set_option('display.max_columns', 50, 'display.width', 200)
df = pd.read_csv('input/Unicorns_Completed.csv')
```

2.2 Data Cleaning

```
import re
def convert_years_months(s):
    m = re.match(r'(\d+)y?\s?(\d+)m?o?', s)
    return f'{m[1]}y{m[2]}m' if m else s

df['Years to Unicorn'] = df['Years to Unicorn'].apply(convert_years_months)

def correct_industry_labels(s):
    if s == 'Health':
        return 'Healthcare & Life Sciences'
    if s == 'West Palm Beach':
        return 'Enterprise Tech'
    return s

df['Industry'] = df['Industry'].apply(correct_industry_labels)
```

2.3 Prepare Data

```
df['Unicorn Date'] = pd.to_datetime(df['Unicorn Date'])
df['Valuation ($B)'] = pd.to_numeric(df['Valuation ($B)'])
df['Unicorn Year'] = df['Unicorn Date'].dt.year
df['Funding ($B)'] = df['Total Equity Funding ($)'] / 1_000_000_000
```

2.4 Preview Data

```
df.head()
```

3 Descriptive Analysis

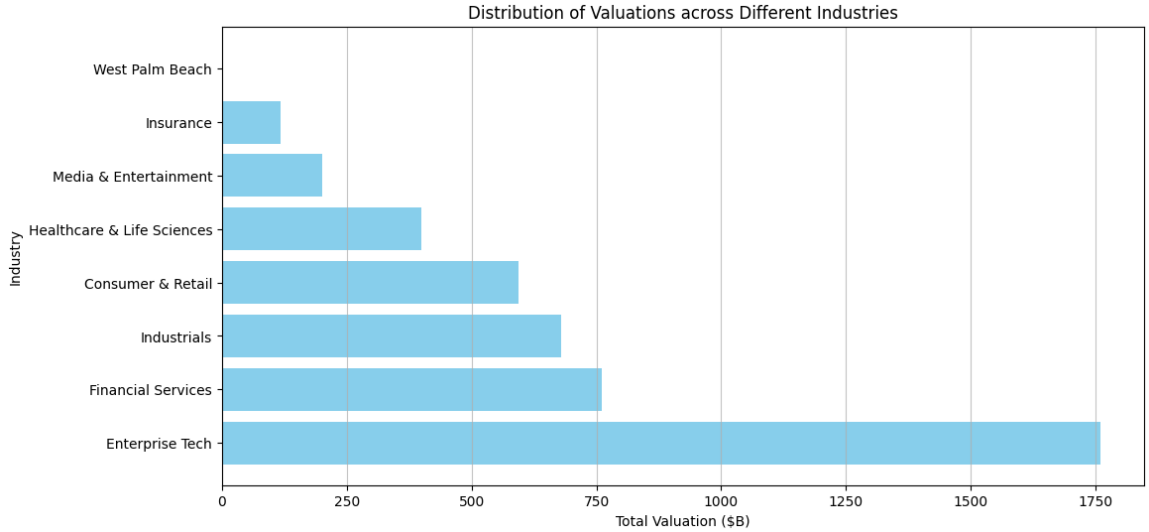
3.1 Valuations

3.1.1 Distribution of Valuations across Different Industries

```
# Group by industry and sum valuations
```

```
industry_valuation_df = df.groupby('Industry')['Valuation
↪  ($B)'].sum().reset_index().sort_values('Valuation ($B)', ascending=False)
industry_valuation_df
```

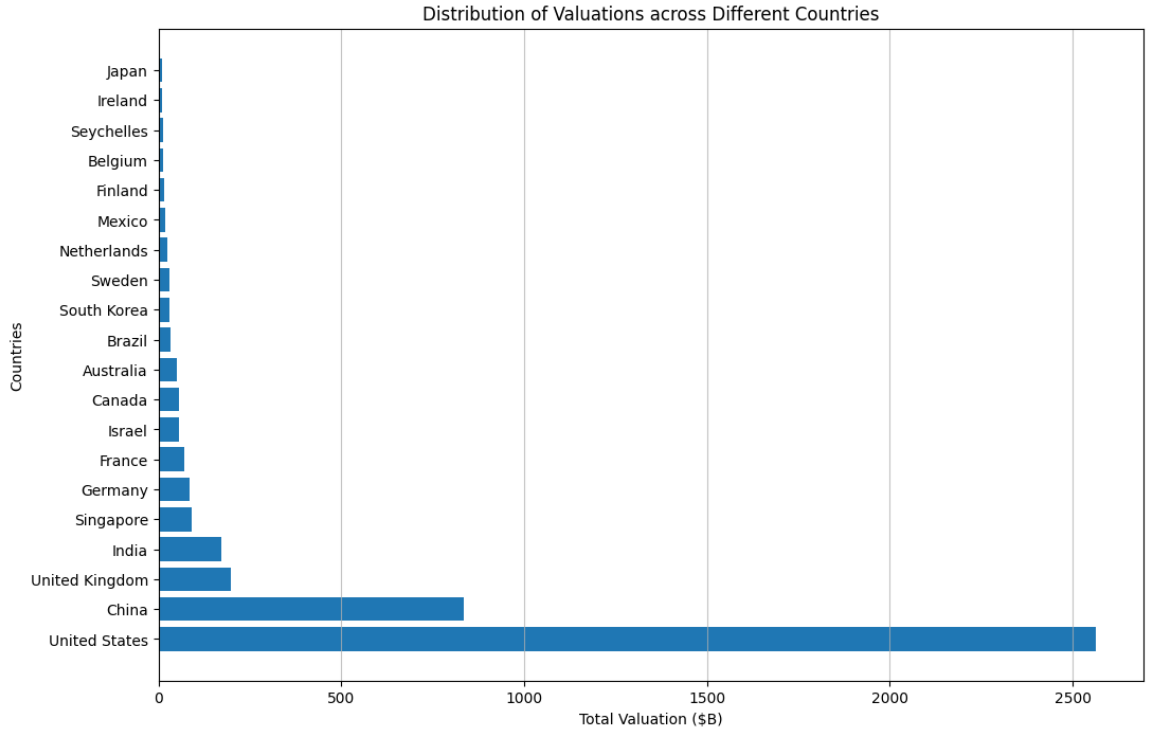
```
plt.figure(figsize=(12, 6))
plt.barh(industry_valuation_df['Industry'], industry_valuation_df['Valuation ($B)'],
↪  color='skyblue')
plt.title('Distribution of Valuations across Different Industries')
plt.xlabel('Total Valuation ($B)')
plt.ylabel('Industry')
plt.grid(axis='x', alpha=0.75)
```



3.1.2 Distribution of Valuations across Different Countries

```
# Group by Country and sum valuations
country_valuation_df = df.groupby('Country')['Valuation
↪  ($B)'].sum().reset_index().sort_values('Valuation ($B)', ascending=False).head(20)
country_valuation_df
```

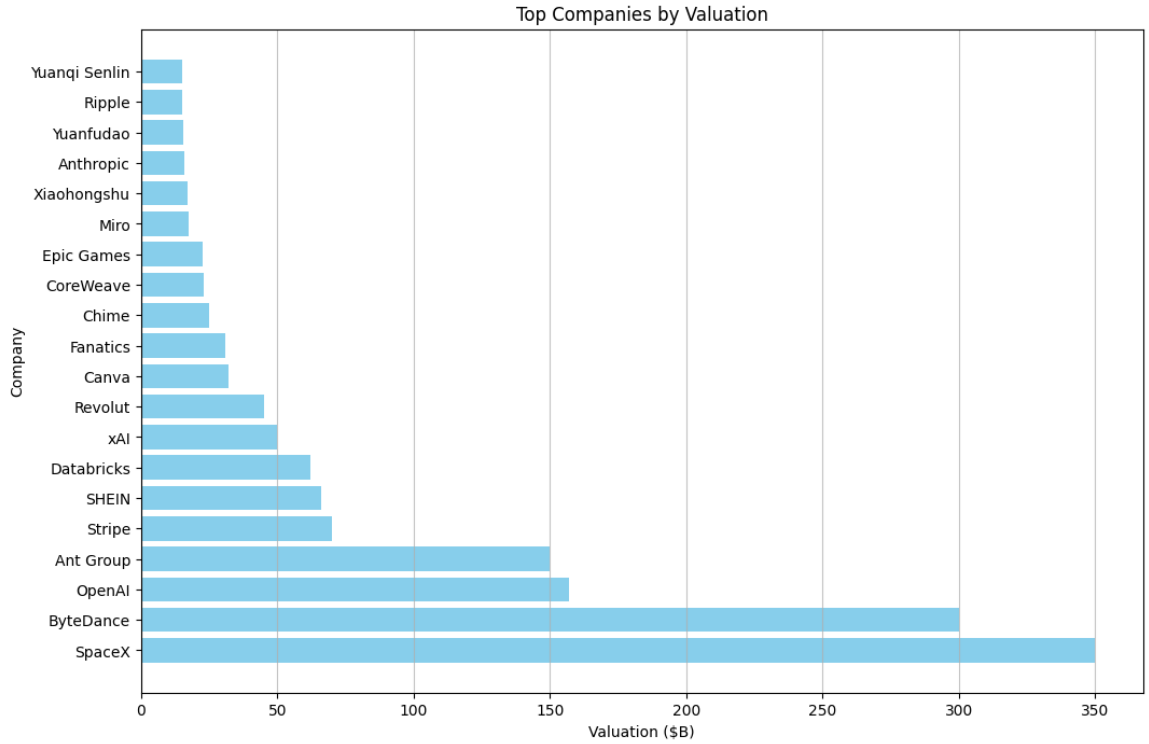
```
plt.figure(figsize=(12, 8))
plt.barh(country_valuation_df['Country'], country_valuation_df['Valuation ($B)'])
plt.title('Distribution of Valuations across Different Countries')
plt.xlabel('Total Valuation ($B)')
plt.ylabel('Countries')
plt.grid(axis='x', alpha=0.75)
plt.show()
```



3.1.3 Top Companies by Valuation

```
top_companies = df.sort_values(by='Valuation ($B)', ascending=False).head(20)
top_companies
```

```
plt.figure(figsize=(12, 8))
plt.barh(top_companies['Company'], top_companies['Valuation ($B)'], color='skyblue')
plt.title('Top Companies by Valuation')
plt.xlabel('Valuation ($B)')
plt.ylabel('Company')
plt.grid(axis='x', alpha=0.75)
plt.show()
```

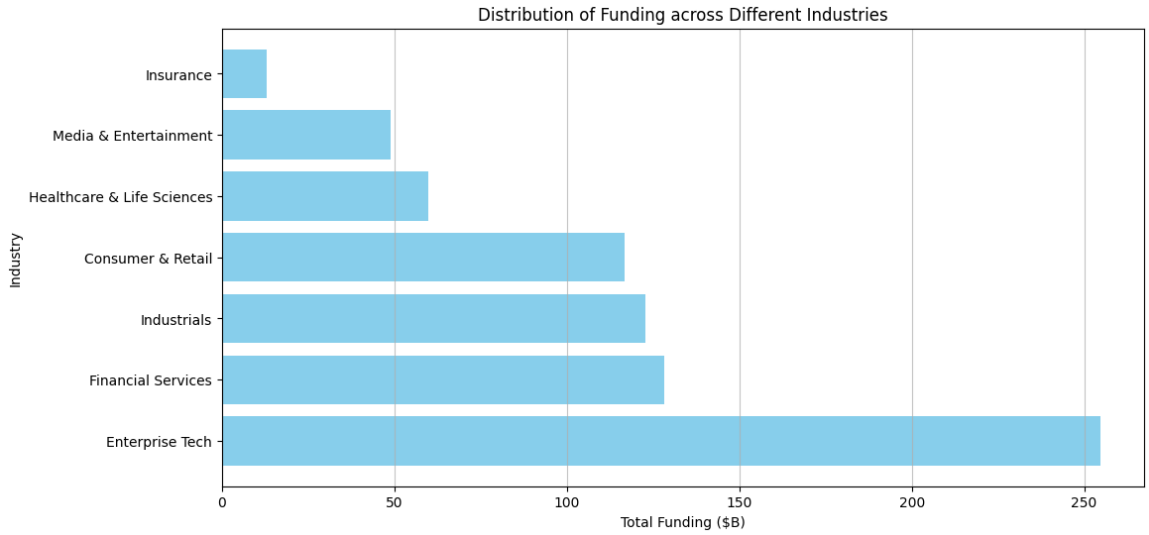


3.2 Funding

3.2.1 Distribution of Funding across Different Industries

```
# Group by industry and sum valuations
industry_funding_df = df.groupby('Industry')['Funding ($B)'].sum().reset_index().sort_values('Funding ($B)', ascending=False)
industry_funding_df
```

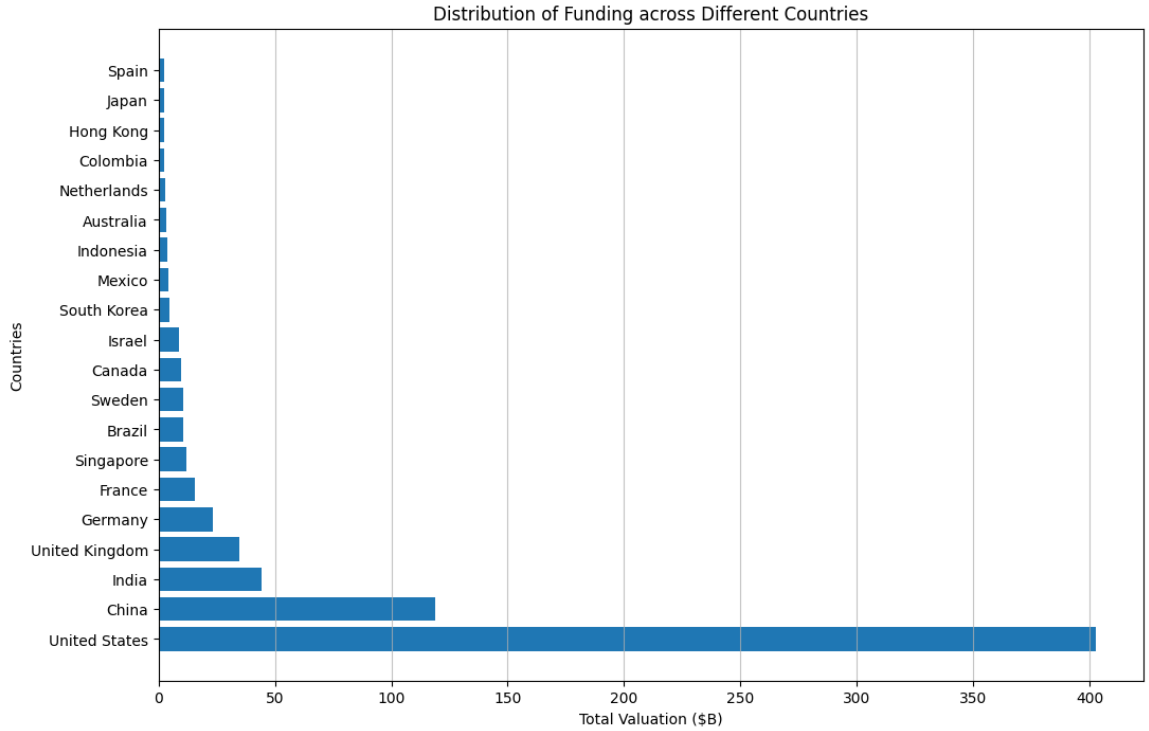
```
plt.figure(figsize=(12, 6))
plt.barh(industry_funding_df['Industry'], industry_funding_df['Funding ($B)'], color='skyblue')
plt.title('Distribution of Funding across Different Industries')
plt.xlabel('Total Funding ($B)')
plt.ylabel('Industry')
plt.grid(axis='x', alpha=0.75)
```



3.2.2 Distribution of Funding across Different Countries

```
# Group by Country and sum valuations
country_funding_df = df.groupby('Country')['Funding
↪  ($B)'].sum().reset_index().sort_values('Funding ($B)', ascending=False).head(20)
country_funding_df
```

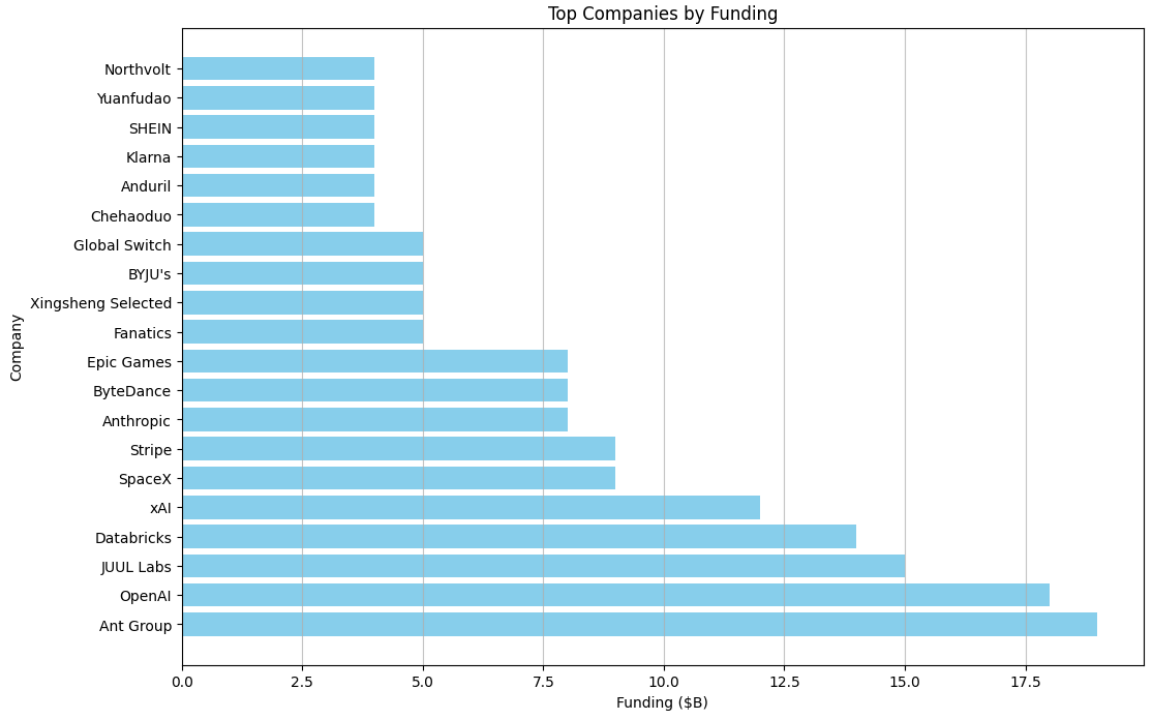
```
plt.figure(figsize=(12, 8))
plt.barh(country_funding_df['Country'], country_funding_df['Funding ($B)'])
plt.title('Distribution of Funding across Different Countries')
plt.xlabel('Total Valuation ($B)')
plt.ylabel('Countries')
plt.grid(axis='x', alpha=0.75)
plt.show()
```



3.2.3 Top Companies by Funding

```
top_companies = df.sort_values(by='Funding ($B)', ascending=False).head(20)
top_companies
```

```
plt.figure(figsize=(12, 8))
plt.barh(top_companies['Company'], top_companies['Funding ($B)'], color='skyblue')
plt.title('Top Companies by Funding')
plt.xlabel('Funding ($B)')
plt.ylabel('Company')
plt.grid(axis='x', alpha=0.75)
plt.show()
```



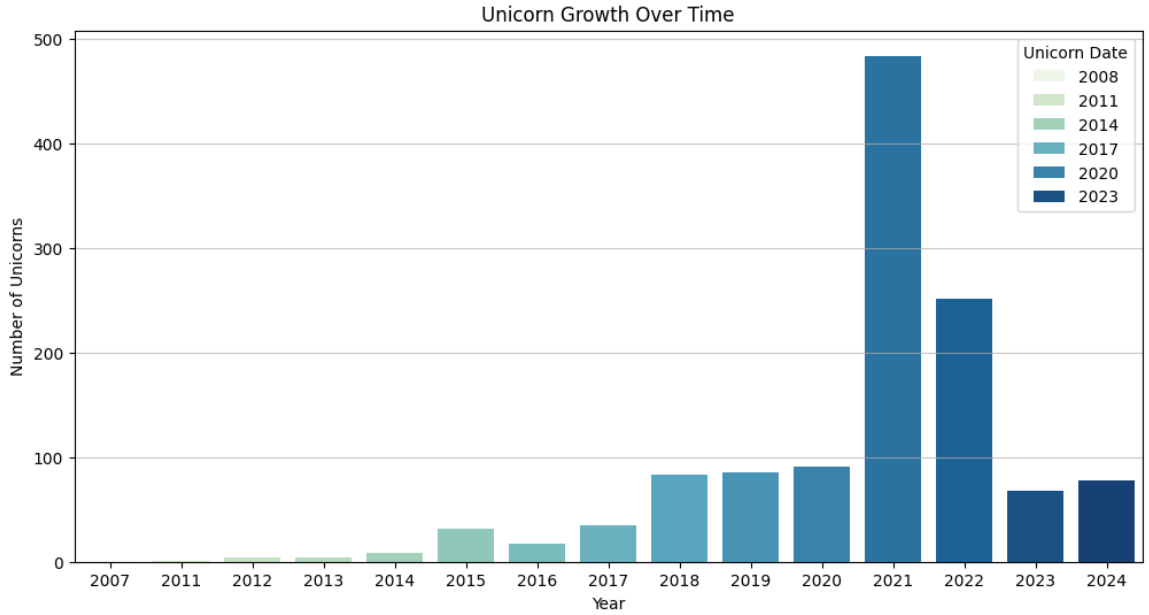
4 Time-Based Analysis

4.1 Unicorn Growth Over Time

```
unicorn_count = df.groupby(df['Unicorn Date'].dt.year).size()
unicorn_count
```

```
Unicorn Date
2007      1
2011      1
2012      4
2013      4
2014      9
2015     32
2016     17
2017     35
2018     83
2019     85
2020     91
2021    484
2022    252
2023     68
2024     78
dtype: int64
```

```
plt.figure(figsize=(12, 6))
sns.barplot(x=unicorn_count.index, y=unicorn_count.values, hue=unicorn_count.index,
↪  palette='GnBu')
plt.title('Unicorn Growth Over Time')
plt.xlabel('Year')
plt.ylabel('Number of Unicorns')
plt.grid(axis='y', alpha=0.7)
plt.show()
```



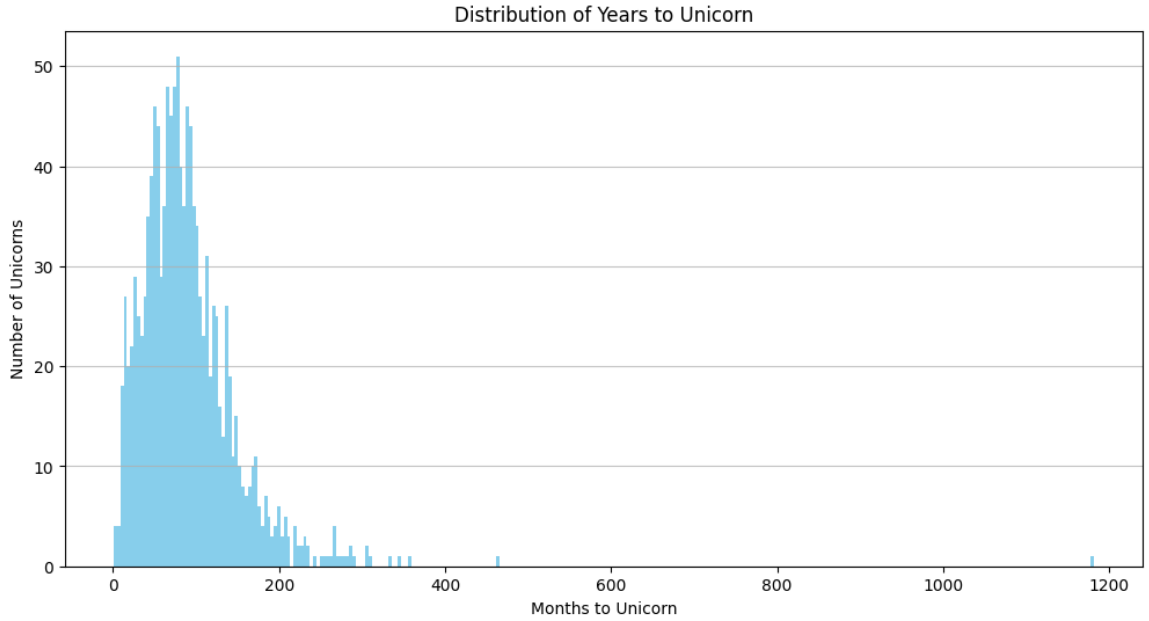
4.2 Time to Unicorn

```
# Function to convert "Years to Unicorn" into total months
def convert_years_to_months(years_str):
    if 'y' in years_str and 'm' in years_str:
        years, months = years_str.split('y')
        months = months.replace('m', '').strip()
        return int(years.strip()) * 12 + int(months)
    elif 'y' in years_str:
        years = years_str.replace('y', '').strip()
        return int(years) * 12
    elif 'm' in years_str:
        months = years_str.replace('mo', '').replace('m', '').strip()
        return int(months)
    else:
        return None

df['Years to Unicorn (Months)'] = df['Years to Unicorn'].apply(convert_years_to_months)
```

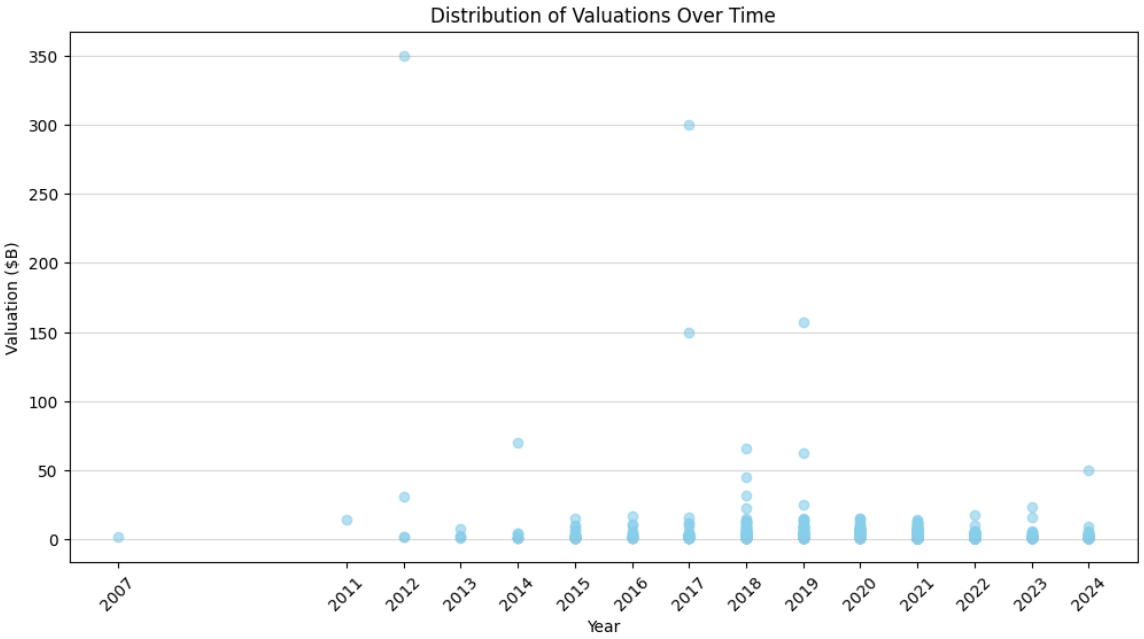
```
plt.figure(figsize=(12, 6))
plt.hist(df['Time to Unicorn (Months)'].dropna(), bins=300, color='skyblue')
plt.title('Distribution of Time to Unicorn')
```

```
plt.xlabel('Months to Unicorn')
plt.ylabel('Number of Unicorns')
plt.grid(axis='y', alpha=0.75)
plt.show()
```



4.3 Distribution of Valuations Over Time

```
plt.figure(figsize=(12, 6))
plt.scatter(df['Unicorn Year'], df['Valuation ($B)'], alpha=0.6, color='skyblue')
plt.title('Distribution of Valuations Over Time')
plt.xlabel('Year')
plt.ylabel('Valuation ($B)')
plt.xticks(df['Unicorn Year'].unique(), rotation=45)
plt.grid(axis='y', alpha=0.5)
plt.show()
```


Distribution of Valuations Over Time
Valuation ($B)
350
300
250
200
150
100
50
0
2007
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
Year